

We claim:

1. A method for revising a computer program, which comprises:

providing a computer program written in a programming language
in a computer;

prescribing consistency, syntax, grammar, and lexical rules;

searching the computer program initially for infringements of
the prescribed consistency, syntax, grammar rules, and lexical
rules;

for an infringement of a prescribed rule, calculating a
possible correction in the computer program; and

changing the computer program in accordance with the
calculated correction.

2. The method according to claim 1, which further comprises,
for an infringement of a prescribed rule, calculating a
plurality of possible corrections of the rule infringement in
the computer program.

3. The method according to claim 2, which further comprises, for an infringement of a prescribed rule, automatically selecting a correction option from the plurality of correction options.

4. The method according to claim 2, which further comprises, for an infringement of a prescribed rule, interactively selecting a correction option from the plurality of correction options.

5. The method according to claim 1, which further comprises: searching the computer program for infringements of prescribed rules as it is gradually input; and

graphically identifying the infringements before the end of input.

6. The method according to claim 1, which further comprises: searching the computer program for infringements of prescribed rules actually as the computer program is gradually input; and

automatically correcting a prescribed type of infringement before the inputting has been completed.

7. A computer program product, comprising:

a computer program for loading directly into an internal memory of a computer, said computer program including computer program code sections for

prescribing consistency, syntax, grammar, and lexical rules;

searching the computer program initially for infringements of the prescribed consistency, syntax, grammar rules, and lexical rules;

for an infringement of a prescribed rule, calculating a possible correction in the computer program; and

changing the computer program in accordance with the calculated correction.

8. A computer program product stored on a medium suitable for computers, the computer program product comprising:

computer-readable program devices permitting a computer to execute a method including the steps:

prescribing consistency, syntax, grammar, and lexical rules;

searching the computer program initially for infringements of the prescribed consistency, syntax, grammar rules, and lexical rules;

for an infringement of a prescribed rule, calculating a possible correction in the computer program; and

changing the computer program in accordance with the calculated correction.

9. A programmed data medium, comprising:

a data medium; and

a computer program permitting a computer to execute a method including:

prescribing consistency, syntax, grammar, and lexical rules;

searching the computer program initially for infringements of the prescribed consistency, syntax, grammar rules, and lexical rules;

for an infringement of a prescribed rule, calculating a possible correction in the computer program; and

changing the computer program in accordance with the calculated correction.

10. A computer system, comprising:

a computer; and

means for executing a method on said computer, the method including:

prescribing consistency, syntax, grammar, and lexical rules;

searching the computer program initially for infringements of the prescribed consistency, syntax, grammar rules, and lexical rules;

for an infringement of a prescribed rule, calculating a possible correction in the computer program; and

changing the computer program in accordance with the calculated correction.

11. A computer system for revising a computer program written in a programming language, the computer system comprising:

a memory device storing a computer program in a storage medium;

a processing unit reading the computer program from the memory device and analyzing the computer program;

said processing unit searching the computer program initially for infringements of prescribed consistency, syntax, grammar, and lexical rules;

for an infringement of a prescribed rule, said processing unit calculating a possible correction in the computer program;

said processing unit changing the computer program in accordance with the calculated correction;

said processing unit then revising the computer program to said memory device; and

an output device reading the revised computer program from said memory device and outputting the revised computer program.

12. The computer system according to claim 11, wherein, for an infringement of a prescribed rule, said processing unit calculates a plurality of possible corrections of the rule infringement in the computer program.

13. The computer system according to claim 12, wherein, for an infringement of a prescribed rule, said processing unit

automatically selects a correction option from the plurality of correction options.

14. The computer system according to claim 12, wherein, for an infringement of a prescribed rule, said processing unit interactively selects a correction option from the plurality of correction options.

15. The computer system according to claim 11, wherein said processing unit searches the computer program for infringements of prescribed rules actually as the computer program is gradually input, and said output device identifies graphically the infringements before the computer program is completely input.

16. The computer system according to claim 11, wherein said processing unit searches the computer program for infringements of prescribed rules as the computer program is gradually input, and automatically corrects a prescribed type of infringement before the computer program is completely input.

17. A method for revising a computer program written in a programming language, which comprises:

providing a computer;

analyzing, with the computer, a computer program for infringements of prescribed consistency, syntax, grammar, and lexical rules; and

defining ignored infringements from the prescribed infringements, the ignored infringements being automatically ignored.

18. The method according to claim 17, which further comprises defining the ignored infringements by virtue of a categorical specification of the infringement.

19. The method according to claim 17, which further comprises defining the ignored infringements by virtue of a generalized specification of the infringement.

20. The method according to claim 17, which further comprises defining the ignored infringements by virtue of a hierarchical specification of the infringement.

21. The method according to claim 17, which further comprises defining the ignored infringements by indicating a declaration environment of the infringement.

22. The method according to claim 17, wherein ignored infringements are defined by specifying an area or context of a construct.

23. The method according to claim 17, wherein ignored infringements are defined by specifying regions of the source code of the computer program, the regions being defined by indicating:

lines and columns;

starting lines and ending lines and starting columns and ending columns;

nodes in a parsing/syntax tree;

starting nodes and ending nodes in the parsing/syntax tree; and

a path in the parsing/syntax tree.

24. The method according to claim 17, which further comprises defining the ignored infringements by indicating a class of constructs.

25. The method according to claim 17, which further comprises defining the ignored infringements by indicating a class of nodes.

26. The method according to claim 25, which further comprises defining ignored infringements by indicating a class of nodes with subnodes.

27. A computer program for loading directly into an internal memory of a computer, the computer program comprising computer program code sections analyzing a further computer program for infringements of prescribed consistency, syntax, grammar, and lexical rules, and defining ignored infringements from the prescribed infringements, the ignored infringements being automatically ignored by the computer program codes sections.

28. A computer program product stored on a medium suitable for computers, the computer program product comprising computer-readable programming means for analyzing a further computer program for infringements of prescribed consistency,

syntax, grammar, and lexical rules, and defining ignored infringements from the prescribed infringements, the ignored infringements being automatically ignored.

29. A programmed data medium, comprising:

a data medium storing a computer program for analyzing a computer program for infringements of prescribed consistency, syntax, grammar, and lexical rules, and defining ignored infringements from the prescribed infringements, the ignored infringements being automatically ignored.

30. A computer system, comprising a computer analyzing a computer program for infringements of prescribed consistency, syntax, grammar, and lexical rules, and defining ignored infringements from the prescribed infringements, the ignored infringements being automatically ignored.

31. A computer system for revising a computer program written in a programming language, comprising:

a memory device for storing a computer program on a storage medium;

a processing unit reading the computer program from said memory device and the computer program, said processing unit searching the computer program initially for infringements of prescribed consistency, syntax, grammar, and lexical rules; and

means for defining infringements, the infringements being automatically ignored during analysis.

32. The computer system according to claim 30, wherein said means for defining infringements define infringements based on categorical specification of the infringement.

33. The computer system according to claim 30, wherein said means for defining infringements define infringements based on generalized specification of the infringement.

34. The computer system according to claim 31, wherein said means for defining infringements define infringements based on hierarchical specification of the infringement.

35. The computer system according to claim 31, wherein said means for defining infringements to be ignored define

infringements by indicating of a declaration environment of the infringement.

36. The computer system according to claim 31, wherein said means for defining infringements to be ignored define infringements specifying an area or context of a construct.

37. The computer system according to claim 31, wherein said means for defining infringements to be ignored define infringements by specifying regions of the source code of the computer program, the regions being defined by indicating:

lines and columns;

starting lines and ending lines and starting columns and ending columns;

nodes in a parsing/syntax tree;

starting nodes and ending nodes in the parsing/syntax tree; and

a path in the parsing/syntax tree.

38. The computer system according to claim 31, including means for defining infringements to be ignored by indicating a class of constructs.

39. The computer system according to claim 31, including means for defining infringements to be ignored by indicating a class of nodes.

40. The computer system as claimed in claim 39, wherein said class of nodes has subnodes.